White Paper

Building Cutting-Edge Server Applications

Intel® Xeon™ Processor Family Features the Intel NetBurst™ Microarchitecture with Hyper-Threading Technology

The Intel® Xeon™ processor family features highly scalable performance, powered by the Intel® NetBurst™ microarchitecture with Hyper-Threading Technology. These technologies allow server applications to support more features, increase transaction throughput and response times, and serve more concurrent users.

Contents

Executive Summary

Bottlenecks. We encounter them all the time — on freeway exits, at the corner coffee shop — and on server racks. Whether you are writing applications for a database server, a Web server, or Web services, end-users' perception of your feature-rich application is affected by the performance of the server itself.

The Intel® Xeon™ processor family features two new processor hardware innovations that help remove processing bottlenecks, increase processing throughput, and provide a significant positive impact on server application performance:

- Intel® NetBurst™ microarchitecture
- Hyper-Threading technology

The Intel NetBurst microarchitecture significantly enhances P6[1] microarchitecture as well as introduces new innovative performance related features, such as Execution Trace Cache, Rapid Execution Engine, Hyper-Pipelined technology, and Streaming SIMD Extensions 2. It also triples the throughput on the system bus, allowing the Intel Xeon processor family to deliver industry-leading performance today and over the next several years. But typically (even with highly optimized code), not all of the available execution resources are used during each clock cycle, due to dependencies, memory latencies and data bottlenecks.

Hyper-Threading technology provides two logical processors on each physical processor and lets applications execute separate tasks on each logical processor. This increases the utilization of resources on the processor, thereby reducing bottlenecks and increasing throughput for multi-threaded applications. A single-threaded application is like a store with one cashier. To move customers through check-out more quickly (higher throughput), the store can add another cashier and additional check-stands. To increase throughput by N, you need N cashiers and N checkstands. This is the situation with a multi-threaded application running on a traditional multi-processor (MP) system. Hyper-Threading technology further extends the parallel processing capabilities of the system. Returning to the store analogy, Hyper-Threading technology is similar to the modern self-scanning check-stand that allows one cashier to supervise multiple check-stands (N cashiers can supervise N*M check-stands). Initial in-house testing has shown that Hyper-Threading technology can provide noticeable performance gains on many multi-threaded server applications.

The Intel Xeon processor family combines Intel NetBurst microarchitecture, with its faster system bus and improved cache, with Hyper-Threading technology. By doing so, the Intel Xeon processor family increases the scalability of multi-threaded server applications by taking advantage of the inherent parallelism for all instructions and the additional parallelism provided by Hyper-Threading technology for multiple threads. The net result is a number of enterprise benefits:

- Increased number of users a server can support
- Improved reaction and response time because twice as many tasks can be executed simultaneously on a multi-processor system
- Increased number of transactions that can be executed
- Compatible with existing IA-32 software

To achieve these benefits, software design must dovetail with hardware capabilities. Applications that are multi-threaded and optimized for these technologies provide leading-edge performance and improved end-user satisfaction. What's more, the investment in optimizing for the Intel Xeon processor family will carry forward across future processor implementations.

To help server application developers quickly and easily optimize their applications for the Intel Xeon processor architecture and meet their early-to-market goals, Intel has developed the necessary tools and compilers along with a development platform. Intel is also working with a range of third-party tool vendors to make application recompilation and porting even easier.

[1] Processors based on the P6 microarchitecture include the Pentium® Pro, Pentium II, Pentium III, and Pentium III Xeon.

Overview of the Intel® Netburst™ Microarchitecture

The Intel NetBurst microarchitecture offers several new technologies and innovative features that result in more efficient code execution over the P6 microarchitecture found in the Pentium® III Xeon™ processor:

Feature Description	Key Benefits for Server Applications
Extremely High Clock Speeds - Attains clock speeds up to 1.6 GHz for MP servers and in excess of 2.0 GHz for DP servers.	Raw execution is faster, allowing for both optimization of existing code and the addition of new application features and enhancements.
400-MHz System Bus - Uses a physical signaling scheme of quad pumping, data transfers over a 100-MHz clocked system bus, and a buffering scheme that allows for sustained 400-MHz data transfers.	This system bus delivers a data rate of 3.2 GB/s in and out of the processor, thereby increasing the throughput of multi-processing and multi-threaded server applications.
Advanced Dynamic Execution - Keeps the execution units busy by using a very deep, out-of-order speculative execution engine. The Intel NetBurst microarchitecture can have up to 126 instructions in flight. The Advanced Dynamic Execution engine also delivers an enhanced branch prediction capability.	The 4K branch target array stores predicted program counters (branch targets) for branches. The large array can store many targets, reducing latencies and branch misprediction by about 33%, in initial in-house testing, over the P6 microarchitecture.
Rapid Execution Engine - Runs Arithmetic Logic Units (ALUs) within the processor at two times the frequency of the processor core.	All applications (legacy and new) take advantage of this feature; no code changes are required. Some of the ALU and logical operations are done in half the time.
Hyper-Pipelined Technology - Doubles the pipeline depth, to a 20-stage pipeline. This technology significantly increases processor performance and frequency scalability of the base microarchitecture.	If the software is written with modern techniques, the application becomes more scalable to the frequency and achieves higher performance.
Execution Trace Cache - First-level cache stores decoded instructions (also known as micro-ops); delivers micro-ops to the processor core at high speed.	In the previous generation of processors, the decoder was part of the execution loop. With the Intel NetBurst microarchitecture, the decoder has been uncoupled from the execution loop by using the Execution Trace Cache. The trace cache delivers micro-ops to the core at a very high bandwidth and is naturally suited to take advantage of instruction-level parallelism in software.
Advanced Transfer Cache - Second-level cache delivers data and instructions to the processor core at high speed with larger cache lines.	Because the Advanced Transfer Cache is clocked at the same rate as the processor core, as faster processors are released this cache speed increases correspond-ingly, providing high-speed access to key data. The larger cache line sizes also decrease cache misses.
Integrated Level 3 Cache - Large on-die cache available on the Intel Xeon processor MP for 4-way and above server platforms.	Additional cache space is available for large instruction and data sets, which significantly reduces memory latency and improves performance for mid-range and high-end server applications.
Streaming SIMD Extensions 2 (SSE2)[2] - Extends the SIMD capabilities of Intel MMX™ technology and the SSE instructions by adding 144 new instructions.	New instructions let programmers execute program tasks with fewer instructions in less time by taking advantage of data parallelism. Intel offers performance libraries to help programmers use SSE2; see "Resources" for more information. SSE2 has been shown to provide significant performance improvements in front-end server applications in security.
More Performance Counters	These counters provide a deeper look at code execution for more efficient application tuning.

[2] To take advantage of this feature, your code must be specifically optimized. See “How to Obtain the Best Performance” and “Resources” for more information.

Overview of Hyper-Threading Technology

Advances in today's processors have historically focused on instruction-level parallelism (ILP) and on higher clock speeds. The goal has been to find more independent instructions to execute during each clock cycle, to achieve higher performance. Some of the new advances in the Intel NetBurst microarchitecture have helped improve instruction-level performance. With the Intel Xeon processor family, Intel has added ground-breaking new technology that goes beyond ILP to exploit thread-level parallelism (TLP).

In the past, exploiting TLP meant deploying symmetrical multi-processor (SMP) servers, where multiple application threads could be run on separate processors within a server. Hyper-Threading technology is a form of simultaneous multi-threading technology (SMT), where multiple applications, or multiple threads of a single application, can be run simultaneously on one processor. The Intel Xeon processor family implementation of Hyper-Threading technology enables each physical processor to appear as two logical processors to the operating system and software.

Each logical processor maintains an independent architectural state, and can respond to interrupts independently. The two logical processors within each physical processor share the physical execution resources. Doing so allows a second, simultaneous thread to use execution resources that are not being used when only one thread is executing. The result is an increased utilization of the execution resources within each physical processor package. This improvement in CPU resource utilization yields higher processing throughput for the application.

Figure 1 compares a processor with Hyper-Threading technology to a processor without Hyper-Threading technology. Hyper-Threading technology is well-suited for multi-processor systems and can further enhance their performance. On a multi-processor platform, the operating system can schedule separate threads to execute not only on each physical processor simultaneously, but on each logical processor simultaneously as well. This improves overall performance and system response because many parallel tasks can be dispatched sooner due to twice as many logical processors being available to the system.

In the past, sharing memory across multiple CPUs could limit the performance scaling, due to bottlenecks in accessing shared memory. Because the caches are shared among the logical processors in a physical processor, Hyper-Threading technology can benefit applications that implement a pair of concurrent threads

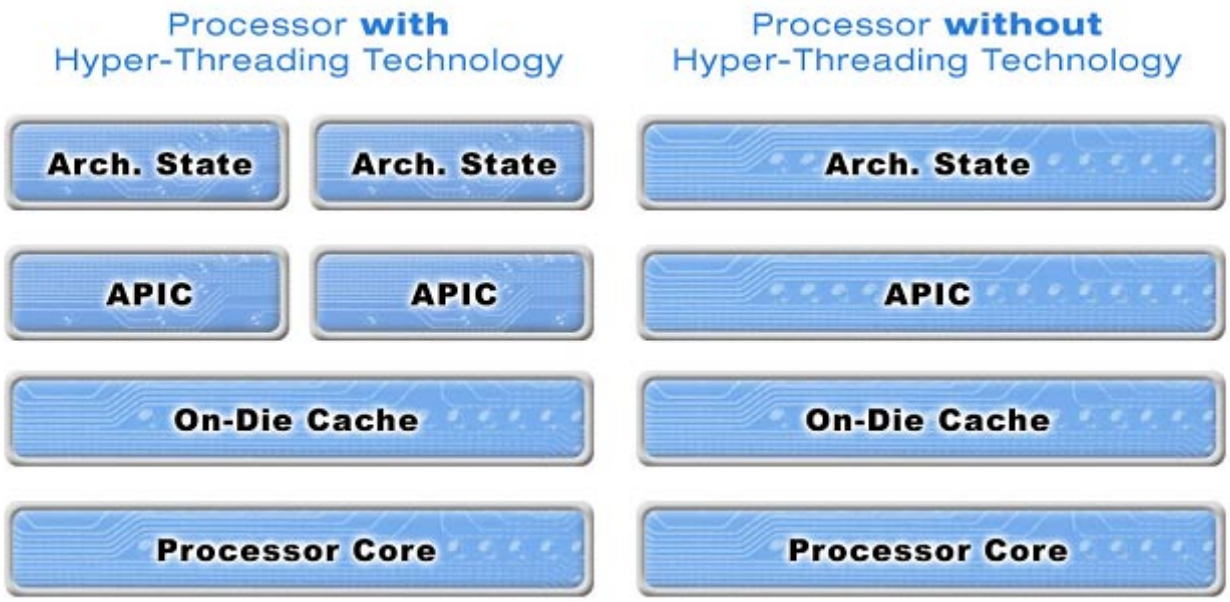

Figure 1. Comparison of an Intel® Xeon™ Processor with Hyper-Threading Technology to an Intel Processor without Hyper-Threading Technology

that produce and consume data. Data can be produced by one thread on one logical processor, and the same data will be readily available to the consumer thread running on the other logical processor within the same physical processor, with no external bus access slowing down the process.

To demonstrate how resource sharing can benefit overall performance, consider a group putting up party decorations. The resources are tape, stapler, tacks, balloons, and crepe paper. If there are several people working, but they all want to use the tape, they must wait for one task to be completed before starting another, and the process is slow and inefficient. In contrast, if they order their tasks so that a tape-intensive task is scheduled while someone else uses the tacks and stapler, the job goes more smoothly and quickly. In just the same manner, if an application that is multi-threaded in such a way that different threads use different resources (such as scheduling integer computation-intensive work on one logical processor while running floating-point computations on the other), it can increase throughput significantly.

Licensing Considerations

Hyper-Threading technology provides simultaneous multithreading in one physical processor, but it is not equivalent to traditional, two-way multi-processing using two physical processors. The performance gain that can be attributed to Hyper-Threading technology is not equivalent to the performance gain provided by multithreading across two physical processors. For this reason, Intel recommends that software developers who rely on per-processor licensing models distinguish between physical and logical processors and count only the number of physical processors when determining license fees.

Operating System Considerations

Currently, there are two main operating environments: Microsoft Windows* and Linux. For Windows, there are three different products deserving some discussion:

1. Windows NT*: There is no support for Hyper-Threading technology in this OS, and some applications, optimized or not, may not see any performance increase. This OS does not recognize the second logical processor and operates as if it were not there.
2. Windows 2000: This OS does provide some support in the scheduler, but the scheduler is not fully optimized. Windows 2000 does not provide a mechanism to distinguish between physical and logical processors, so the OS will report a DP system as a 4-processor system. There might be some licensing issues that you may need to address, depending on the system configuration and OS version.
3. Windows .NET* Server: This is the preferred OS on which to implement hyper-threaded technology optimizations in your applications.This OS is optimized for the Intel Xeon processor family and Hyper-Threading technology and is expected to provide the best performance. It also provides a mechanism to distinguish between physical and logical processors.

The Linux 2.4 kernel does not support Hyper-Threading technology today. However, a patch is available that provides support for Hyper-Threading technology. A new release of Linux 7.1, when available, is expected to support Hyper-Threading technology and to include additional kernel-level optimizations.

Server platform vendors may include a method in the BIOS that provides the ability to enable or disable Hyper-Threading technology for software development and tuning purposes.[3]

Performance Improvements and Scalability

The Intel Xeon processor family offers a powerful combination with the Intel NetBurst microarchitecture and Hyper-Threading technology. The Intel NetBurst microarchitecture can exploit instruction level parallelism and executes instructions at a very high clock rate to deliver high performance, while Hyper-Threading technology can take advantage of task-level parallelism and opens a huge window of opportunity in multi-threaded application performance.

[3] When Hyper-Threading technology is disabled, the BIOS tells the operating system to ignore the second logical processor; and the execution resources in each physical processor are fully available to software.

Software that is optimized to run on the Intel Xeon processor family can achieve scalable performance in three ways: [4]

- Processor frequency scaling
- Multi-processor scaling
- Scaling due to Hyper-Threading technology

Let's look at two typical scenarios, to see how the Intel NetBurst microarchitecture with Hyper-Threading technology might benefit real-world server applications.

- **Database Servers**: The higher bandwidth system bus (3.2 GB/s) allows multi-processor systems to provide greater throughput and reduce bottlenecks. More logical processors can run independent tasks, enabling servers to better handle complex queries, concurrent background processes, and concurrent server processes. More database transactions can be processed, providing greater processing power, faster response times, or support for additional concurrent users.
- **Web Servers:** End-users spend less time waiting unproductively. Consider an end-user ordering some books over the Web. This single transaction can span multiple server tasks. A few of these tasks might be creating an invoice and updating inventory. In this scenario, one logical processor can handle invoicing while the other logical processor handles updating inventory. And the overall I/O enhancements at the system level deliver more data faster.

Many other areas stand to gain from the Intel Xeon processor family with Intel NetBurst microarchitecture and Hyper-Threading technology. From supply chain management to voice/computer telephony integration to customer resource management, they all may show similar increased throughput, improved end-user experience, and increased ability to support new product features.

How to Obtain the Best Performance

Existing application code runs correctly on the Intel Xeon processor family. However, as Figure 2 shows, different levels of optimization are possible.[5] To obtain the optimum benefit from the new technology, Intel recommends that you perform some relatively simple code modifications. The numbers in Figure 2 correspond to the following steps, which will help you optimize your code. Tools and support from Intel will assist you through the process:

1. **Benchmark** against the Intel Pentium III Xeon processor. Identify a reproducible workload that your application uses to characterize the existing application performance. Use this same benchmark to gauge progress as you make changes in support of the Intel NetBurst microarchitecture and Hyper-Threading technology.
2. **Recompile** with an Intel NetBurst microarchitecture-optimized compiler, such as the Intel C/C++ and Fortran Compilers, version 5.0 and later, or Microsoft Visual Studio* .NET.
3. **Tune your software architecture to the Intel Xeon processor-based platform**. This involves examining two separate areas:

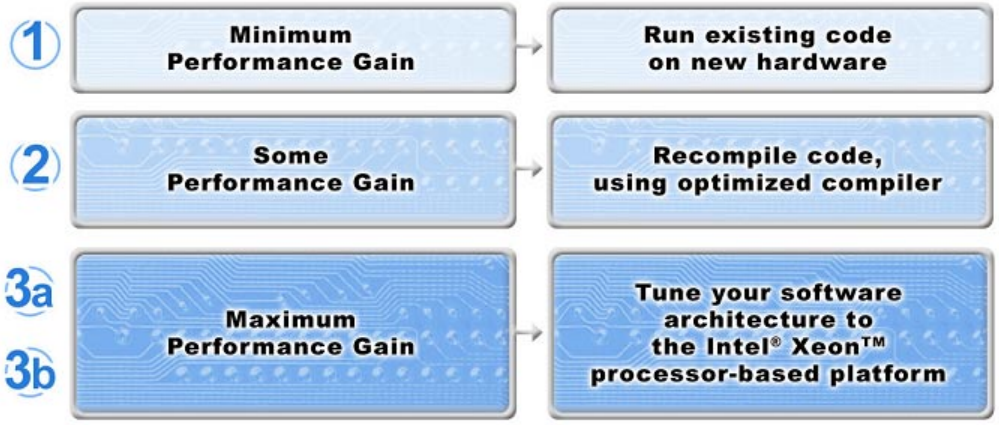

Figure 2. Obtaining the Optimum Performance from Your Application

[4] In any MP system, the scalability of performance is highly dependent on the nature of the application and threading model.
[5] In general, code optimized for Intel NetBurst microarchitecture with Hyper-Threading technology is backwardly compatible to other IA-32 architectures. SSE2 is not backwardly compatible with previous architectures. Two binaries, or dual code paths, are required for backward compatibility if you are implementing SSE2.

Is your application already multi-threaded? Multi-thread applications in areas where you can split the workload across processors. Identify workloads with independent functionalities or data requirements, and create multiple threads to take advantage of all logical processors in the platform. Multi-threading can be done by hand, using native OS threading support, or with a set of high-level OpenMP* directives. OpenMP support is available with the Intel C/C++ and Fortran Compilers 5.0 or later.

3b **Is your application optimized to take advantage of the Intel NetBurst microarchitecture?** Tune the application for common workloads to achieve optimal performance. Consider using the Intel performance libraries, adding SSE2 support, and using the Intel VTune™ Performance Analyzer to identify coding hotspots that hinder good frequency scaling and higher instruction throughput.

For single-threaded code: Apply the techniques covered in Chapter 2 of the *Intel® Pentium® 4 and Intel® Xeon™ Processor Optimization Manual* to achieve efficient frequency scaling and high instruction throughput.

For multi-threaded code: Apply proven techniques, including the following:

- Minimize cost of thread synchronization
- Use PAUSE instruction in spin loops
- Halt idle threads
- Pipeline spin locks to prevent contentions
- Minimize data movement paths

Note: All Intel software development products are available at http://developer.intel.com/software/products/. Useful documents and other Web sites are listed in the “Resources” section at the end of this white paper.

Summary

Writing code that takes advantage of Intel NetBurst microarchitecture with Hyper-Threading technology will benefit server application ISVs by enhancing application scalability and improving application response time, increasing the number of completed transactions, and increasing the number of users served.

The products available at http://developer.intel.com/software/products/, the information available through Intel Software Network http://www.intel.com/software/ and the work Intel is doing with third-party vendors will help ISVs optimize their applications for the Intel NetBurst microarchitecture with Hyper-Threading technology. What’s more, the effort involved in this optimizing process is not a one-time effort that will be outdated when the next innovation comes along. Rather, it is a permanent investment in increased performance that will continue to pay off as hardware innovation continues.

To find out more about optimizing your server applications for Intel NetBurst microarchitecture with Hyper-Threading technology, be sure to visit the sites listed in “More Resources,” and contact your Intel representative.

Resources

Intel NetBurst Microarchitecture Resources

- *Inside the Intel® NetBurst™ Microarchitecture* – http://www.intel.com/eBusiness/pdf/prod/workstation/midhigh/ds011402.pdf
- Intel® Pentium® 4 and Intel Xeon™ processor manuals – http://developer.intel.com/design/pentium4/manuals/

Hyper-Threading Technology Resources

- Hyper-Threading Technology Home Page, a collection of white papers and presentations – http://developer.intel.com/technology/hyperthread/index.htm
- *Introducing Hyper-Threading Technology* – http://developer.intel.com/technology/hyperthread/download/25000802.pdf
- *Detecting Support for Hyper-Threading Technology Enabled Processors* – http://www.intel.com/cd/ids/developer/asmo-na/eng/dc/threading/implementation/20416.htm. This paper is available on the Intel Software Network site.
- *Intel® Pentium® 4 and Intel® Xeon™ Processor Optimization Manual* – This manual will include Hyper-Threading technology optimization information, and will be available on the Intel Developer site – http://developer.intel.com.
- *Intel® Processor Identification and the CPUID Instruction* – Application Note AP-485, Order Number 241618-016 – http://developer.intel.com/design/litcentr/index.htm
- *Using Spin-Loops on Intel® Pentium® 4 Processor and Intel® Xeon™ Processor* – Application Note AP-949 – http://developer.intel.com/design/litcentr/index.htm
- *IA-32 Intel® Architecture Software Developer's Manual* – 3-volume set. Order Numbers 245470, 245471, 245472 – http://developer.intel.com/design/litcentr/index.htm

Other Useful Resources

- Intel® Software Development Products – http:// developer.intel.com/software/products/
- Intel® Developer Site – http://developer.intel.com. Get the latest information for developers on platforms, tools, etc.
- Intel® Developer Services – Personalized developer information, http://developer.intel.com/ids
- "Exploiting Parallelism Using Intel Compiler" – Software Tools and Optimization Track, IDF, Fall 2001.
- "Multithreaded Programming with OpenMP*" – Workstation and Technical Computing Track, IDF, Fall 2001.
- Win32 API documentation – Available to MSDN subscribers at http://premium.microsoft.com/msdn/library

Multi-Threading Resources

If your code is not already multi-threaded, you may want to research multi-threading techniques. A good starting point is *An Introduction to Programming with Threads*, by Andrew Birrell. This is available at http://gatekeeper.dec.com/pub/DEC/SRC/research-reports/abstracts/src-rr-035.html. You can also refer to "Intel and KAI: Creating Tools to Make OpenMP* the Standard for Threading Application Software," available at http://www.intel.com/software/products/compilers/c50/openmp.htm.